

Integration von Datenmodellen

Eine Methodik zum Produktdatenaustausch

Stefan Pfennigschmidt, Peter Kolbe, Peter Jan Pahl

1 Einführung

Das Konzept zur Integration von Datenmodellen ist im Projekt "Verteiltes Facility Management in Telekommunikationsnetzen" (Kurztitel: *FATIMA*) in Kooperation mit der HOCHTIEF Software GmbH erarbeitet worden. *FATIMA* ist ein Projekt im Rahmen des F&E-Programms der DeTeBerkom GmbH, einem Tochterunternehmen der Deutschen Telekom AG.

1.1 Modellbildung

Um die reale Welt in ihrer Komplexität begreifbar und damit nutzbar und veränderbar zu machen, bedient sich der Mensch der Methodik der Modellbildung. Modelle beinhalten dabei die für einen speziellen Sachverhalt notwendigen Informationen über Objekte und Veränderungen der Realität. Indem zwischen - im jeweiligen Kontext - relevanten und vernachlässigbaren Eigenschaften von Objekten unterschieden wird, lassen sich abstrakte Sichten auf die Realität bilden, die durch ihre Anpassung an den speziellen Problembereich eine geringere Komplexität aufweisen und damit handhabbar werden.

Die Abstraktion eines Teilbereiches der Realität beruht auf seiner Gliederung in trennbare Objekte, der eindeutigen Bezeichnung dieser Objekte, und der Klassifizierung der Objekte nach ihren Eigenschaften. Die Objekte bilden dann die Grundlage der Modellierung.

1.2 Begriffe

Klasse: Eine Klasse beschreibt die Eigenschaften gleichartiger Objekte. Für jede relevante Objekteigenschaft enthält sie eine entsprechende Attributdefinition. Die Begriffe Klasse und Objekttyp werden synonym verwandt.

Attribut: Ein Attribut unterscheidet sich von einem Objekt darin, daß es nur als Eigenschaft eines bestimmten Objektes existiert und nicht unabhängig von diesem identifiziert werden kann. Bei der Definition eines Objekttyps wird vereinbart, welche Attribute ein Objekt dieses Typs besitzen kann. Dazu werden Attributname, Typ, Dimension, Grenzen und eventuelle Vorgabewerte angegeben, die dieses Attribut beschreiben. Eine solche Definition wird mit **Attributdefinition** bezeichnet.

Instanz: Ein Objekt wird geprägt, indem ein Objekttyp gewählt und für einen Teil der Attribute Werte festgelegt werden. Es ist nicht erforderlich, daß für alle Attributdefinitionen Belegungen vorgenommen werden. Für einen Objekttyp kann eine beliebige Menge von Objekten geprägt werden. Das Objekt heißt eine Instanz oder Prägung des Objekttyps.

Schema: Eine Menge von Objekttypen, die einen zusammenhängenden Teilbereich der Realität beschreiben, heißt Schema oder Modelltyp.

Modellausprägung: Eine Instanz eines Modelltyps heißt Modellausprägung. Eine Modellausprägung entsteht, indem für eine Teilmenge der Objekttypen des Modelltyps Objekte instanziiert werden.

Semantische Äquivalenz: Durch Zerlegung komplexer Sachverhalte und der damit verbundenen Bildung unterschiedlicher Abstraktionen entstehen vielfach Modelltypen, die sich in Teilbereichen überschneiden. So ist es der Regelfall, daß Eigenschaften eines Objekts in mehreren Modellen eine Rolle spielen. Es existieren demnach Attribute innerhalb der unterschiedlichen Modelltypen, die die gleiche Eigenschaft eines Objekts widerspiegeln, die also die gleiche Bedeutung besitzen. Solche Attribute werden **semantisch äquivalent** genannt.

2 Modellintegration

2.1 Allgemeine Betrachtungen

Der Integrationsprozeß von Modellen in einen Modellverbund beruht auf dem Erkennen semantischer Äquivalenzen zwischen den zu integrierenden Modellen. Äquivalente Informationen können dabei in verschiedenen Modellen unterschiedlich repräsentiert sein. Diese Unterschiede in Repräsentation und Struktur müssen abgeglichen werden. Traditionelle Ansätze verwenden hierzu Konvertierungsmethoden, die die Informationen von einer Struktur (Quellschema) in die andere Struktur (Zielschema) überführen.

Dabei wird tatsächlich nur die Struktur der ursprünglichen Information geändert. Die Information selbst ändert sich nicht. Sie nimmt lediglich eine andere Gestalt an. (Bild 1)

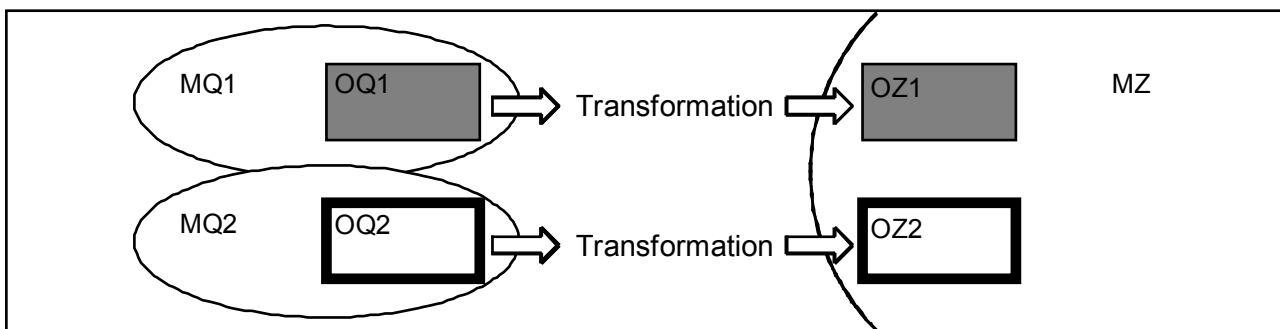


Bild 1: Transformation: Objekte werden ineinander überführt

Wenn wir ein Objekt als Träger von Informationen betrachten, bedeutet das folgendes. Das Objekt OQ1 des Modells MQ1 mit dem Schema SQ1 wird in ein Objekt OZ1 des Modells MZ mit dem Schema SZ umgewandelt. Gleiches gilt für das Objekt OQ2. Man ist mit Hilfe der Transformation demnach in der Lage **Objekte ineinander zu überführen**.

Die Aufgabe des Modellverbundes ist jedoch, gleichbedeutende Informationen zu verbinden. Das heißt, es werden Mechanismen benötigt, die über ein einfaches Transformieren hinausgehen. Man muß in der Lage sein, die Zusammengehörigkeit zweier unterschiedlich strukturierter Informationen

zu erkennen, die nicht durch Transformation auseinander hervorgegangen sind. Auf dieser Basis lassen sich Objekte unterschiedlicher Modelle miteinander verbinden. Objekte werden also nicht mehr nur ineinander überführt, sondern **zusammengeführt**. (Bild 2)

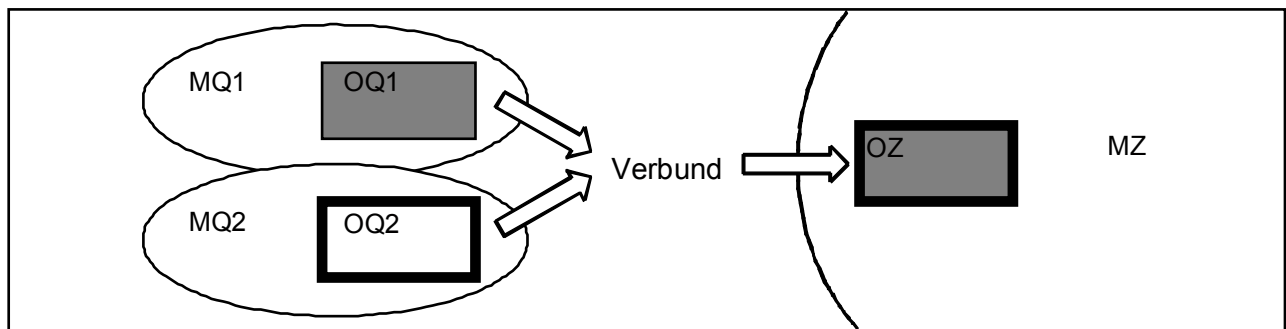


Bild 2: Verbund: Objekte werden zusammengeführt

Aus dem obigen Beispiel wird: Das Objekt OQ1 des Modells MQ1 und das Objekt OQ2 des Modells MQ2 werden aufgrund semantisch äquivalenter Informationen als einander zugehörig erkannt und bilden zusammen das Objekt OZ des Modells MZ.

2.2 Ziel des Modellverbundes

- Modelle werden zu einem logischen Datenbestand zusammengefaßt, der jedoch physisch verteilt sein kann.
- Die Modellausprägungen bleiben auch nach der Integration in ihrer Form erhalten, um sie bei Bedarf wieder extrahieren zu können.
- Anwendungsprogramme können unter Benutzung ihrer ursprünglichen Schemata auf den logischen Datenbestand zugreifen. Ihr Funktionsumfang bleibt dadurch erhalten. Der Informationsaustausch zwischen diesen Anwendungen wird ohne deren Änderung ermöglicht.
- Anwendungsprogramme können unter Benutzung generischer Zugriffe den logischen Datenbestand in einer Gesamtsicht editierbar machen.

2.3 Voraussetzungen

Die Umsetzung einer solchen Integration ist an einige Voraussetzungen gebunden.

Schema: Um Modelle auf semantische Äquivalenzen hin analysieren zu können, ist die Kenntnis der sie beschreibenden Schemainformationen notwendig. Zur Beschreibung dieser Schemata wird die konzeptuelle Beschreibungssprache EXPRESS (ISO 10303–11) verwendet.

Datenzugriff: Eine weitere Voraussetzung ist die Bekanntgabe der Datenzugriffe auf das jeweilig zu integrierende Modell, so daß es auch unabhängig von einer speziellen Anwendung bearbeitet werden kann. Diese Bedingung kann durch Verwendung einer standardisierten Zugriffsschnittstelle realisiert werden. Die ISO 10303-22 definiert das Standard Data Access Interface (SDAI) als eine Menge von Zugriffsfunktionen. Das innerhalb des Projekts entwickelte Laufzeitsystem benutzt dieses Interface.

Objektidentifikation: Bei der dauerhaften Zusammenführung von Objekten verschiedener Modelle ist die Wiederholbarkeit der Objektidentifikation von fundamentaler Bedeutung. Die Integrierbarkeit eines Modells ist dabei von der Eindeutigkeit und Persistenz der Identifikatoren abhängig. Diese beiden Kriterien haben entscheidenden Einfluß auf die Eindeutigkeit und Persistenz der Objektzuordnungen zwischen den Modellen. Ein weiteres Problem, das von der Art und Weise der Objektidentifikation zwischen den Modellen abhängt, ist das Problem der Änderung und

Erzeugung von Objekten innerhalb des Verbundes. Um die Konsistenz jedes einzelnen Modells gewährleisten zu können, müssen bei der Änderung und Erzeugung von Objekten die innerhalb der Modelle vorgeschriebenen Identifikationsregeln eingehalten werden. Daraus resultiert die Frage, ob nur mit der Kenntnis der Objekteigenschaften in einem Modell eine konsistente und eindeutige Erzeugung von Objekten in einem anderen Modell möglich ist.

2.4 Umsetzung

Eine Information ist durch ihre Struktur und ihre Lokalität gekennzeichnet. Das Zusammenführen verschiedener Informationen geschieht demzufolge auf zwei Ebenen. Zum einen müssen die Strukturinformationen (Schemaebene) und zum anderen die Lokalisierungsinformationen (Instanzebene) in Beziehung zueinander gesetzt werden. Daraus ergeben sich folgende Zuordnungen:

Abgleich auf Schemaebene: (*Schema mapping*)

- in Beziehung setzen der Klassen
- in Beziehung setzen der Attribute

Abgleich auf Instanzebene: (*Object matching*)

- In Beziehung setzen der Objektidentifikationen

2.5 Abgleich der Schemata (*Schema mapping*)

Um die strukturellen Unterschiede zwischen Informationsmengen (Datenmodelle) zu überwinden, müssen die Schemata dieser Modelle in Beziehung zueinander gesetzt werden. Dabei treten aufgrund dieser Unterschiede Konflikte auf. Das sind beispielsweise strukturelle Konflikte, wie unterschiedliche Klassenstrukturen, die Einschränkung von Klassen und Attributen oder der Einfluß von Klassen und Attributen bzw. Konflikte, die durch Unterschiede in der Repräsentation von Einzelinformationen auftreten können, wie z.B. Unterschiede bei Einheiten oder im Genauigkeitsgrad. In [3] ist eine vollständige Klassifikation der möglichen Konflikte enthalten.

Zur Lösung dieser Konflikte wurden spezielle Strukturen (*Mapping model*) entwickelt. Das Schema dieses *Mapping models* ist ebenfalls mit EXPRESS spezifiziert. Der Zugriff auf die Abbildungsinformationen kann demzufolge mittels des auf SDAI basierenden Laufzeitsystems erfolgen.

Die grundlegende Struktur ist in Bild 3 dargestellt. Zur Lösung der Konflikte auf Klassenebene wird eine *class_map* aufgebaut, welche die Klassen der Quellschemata mit den Klassen des Zielschemas in Beziehung setzt. Dabei können Bedingungen angegeben werden, unter denen die Klassenbeziehung gültig ist.

Zur Abbildung von Unterschieden auf Attributebene werden Konvertierungsfunktionen (*map_functions*) eingesetzt. Die Informationen, die in den Quellschemata enthalten sind (Attribute und/oder Konstanten), gehen dabei als Parameter in den Funktionsaufruf ein.

Die Beziehungen zwischen Klassen und ihren Attributen können bidirektional angegeben werden. Wenn ein Schema als Zielschema und das andere als Quellschema betrachtet wird, beschreiben die Lesefunktionen (*get_functions*) die Überführung der Objekte in Richtung des Zielschemas und die Schreibfunktionen (*put_functions*) die Überführung in Richtung des Quellschemas. Eine solche Vorgehensweise wird Zwei-Wege-Mapping genannt.

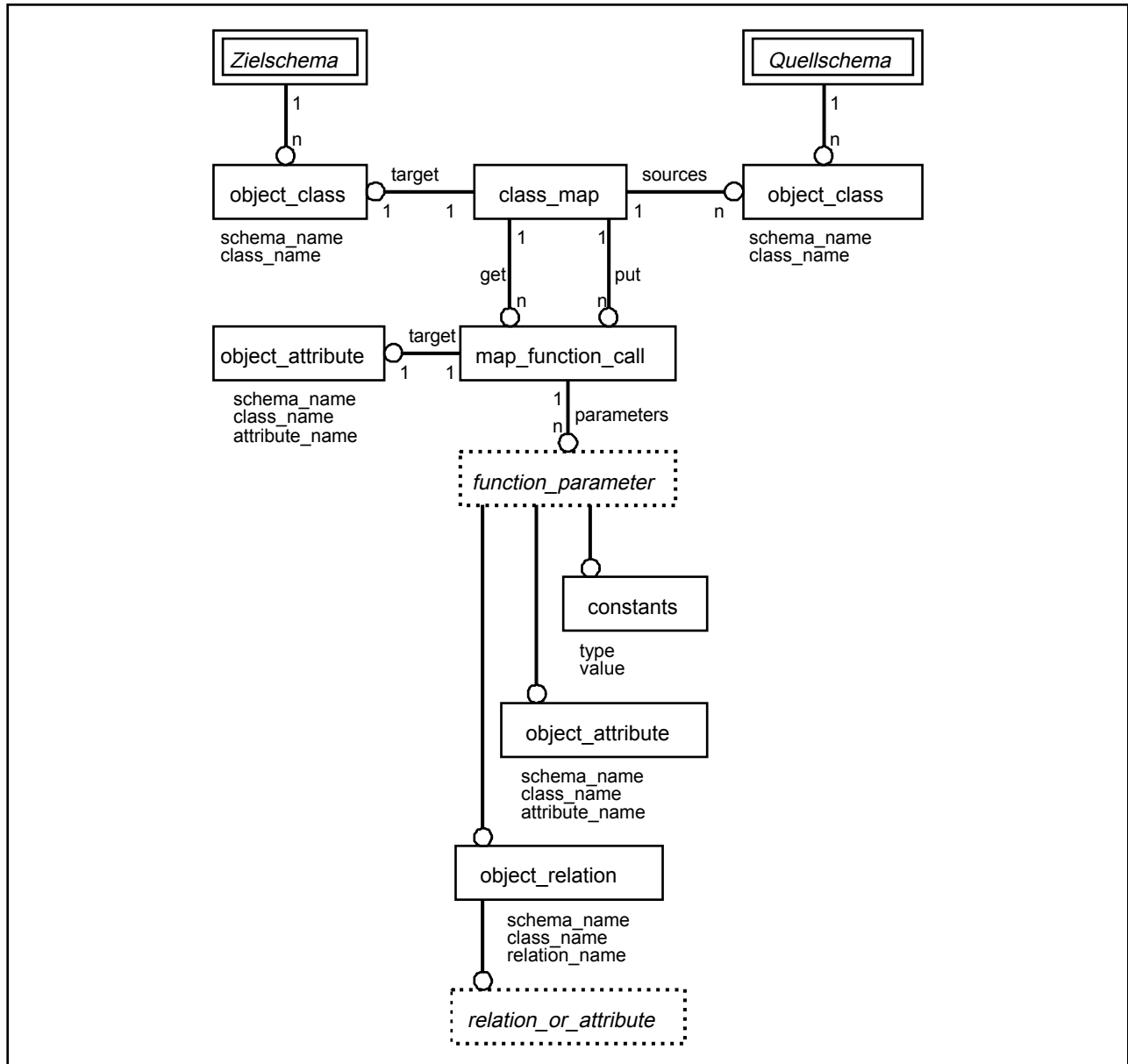


Bild 3: Schematische Darstellung des *Mapping models*

2.6 Abgleich auf Instanzebene (*Object matching*)

Der zweite Integrationsschritt ist die Zuordnung von Objekten mit semantisch äquivalenten Informationen. Dabei soll einem Objekt (Zielobjekt) des einen Modells (Zielmodell) ein entsprechendes Objekt (Quellobjekt) eines anderen Modells (Quellmodell) zugeordnet werden. Der Ansatz zur Lösung dieses Problems ist die Aufstellung einer Zuordnungsregel, die auf Klassenbasis definiert werden kann. Eine solche Regel kann man sich als parametrisierte Datenbankanfrage (*Query*) vorstellen. In einer Anfragesprache wie SQL könnte eine solche parametrisierte Anfrage folgendermaßen aussehen:

```
SELECT * FROM klasse WHERE (attr1 = para1) & (attr2 = para2)
```

Zum Auffinden eines konkreten Objektes des Quellmodells werden die jeweiligen Parameter dieser Anfrage durch aktuelle Werte ersetzt und so eine vollständige Anfrage erzeugt. Die aktuellen Parameterwerte müssen mit Kenntnis des Zielobjekts und seiner Umgebung gesetzt werden. Zur Objektumgebung gehören dabei die Beziehungen zu anderen Zielobjekten und zusätzliche

Klasseninformationen. Die Beziehung von formalen Parametern (im Quellmodell) zu den aktuellen Parametern (des Zielmodells) definiert die Zuordnungsregel.

Eine solche Zuordnungsregel besteht demnach aus drei Teilen:

- formale Parameter
- aktuelle Parameter
- Relation zwischen formalen und aktuellen Parametern

Definition von Zuordnungsregeln: Eine Zuordnungsregel stellt die Beziehung zwischen Objekten zweier Klassen dar. Wie oben dargestellt benötigt man dabei einen Ausdruck auf der Seite der Zielklasse (aktuelle Parameter) und einen Ausdruck auf der Seite der Quellklasse (formale Parameter), die durch einen relationalen Operator miteinander verbunden werden. Die allgemeine Form der Zuordnungsregel sieht folgendermaßen aus.

```
boolean_value := target_expression relational_operator source_expression
```

Eine solche Regel kann in funktionaler Notation geschrieben werden, wobei der relationale Operator als Funktionsname und die beiden Ausdrücke als Parameter fungieren.

```
boolean_value := relational_operator(target_expression, source_expression)
```

Die Ausdrücke, die auf Seiten der Ziel- und Quellklasse gebildet werden müssen, entsprechen den Ausdrücken, wie sie zur Abbildung der unterschiedlichen Klassenstruktur benutzt werden. Es bietet sich daher an, die Struktur der *map_function* zu benutzen, um die Zuordnungsregeln abzubilden.

3 Ergebnis

Das Konzept der Integration von Datenmodellen ist in der Lage, Informationen, die verteilt in beliebig strukturierten Modellausprägungen liegen, zusammenzufassen. Das Erkennen semantisch äquivalenter Informationen innerhalb einzelner Modelle bildet die Grundlage für einen Modellverbund. Die Voraussetzungen für die Integrierbarkeit eines Modells sind die explizite Beschreibung seines Schemas und die Verfügbarkeit von Zugriffsfunktionen. Die eindeutige und persistente Identifikation der Objekte innerhalb eines jeden Modells über Attribute bildet die Voraussetzung für das Erkennen der semantischen Äquivalenz zwischen Objekten des Modellverbundes. Der entscheidende Vorteil, den ein solches Integrationskonzept bietet, ist die Möglichkeit, die im Verbund gesammelten Informationen in ihrem Zusammenhang bearbeiten zu können. Die Grenzen zwischen den Datenmodellen werden aufgehoben und die Zusammenhänge der spezifischen Eigenschaften einzelner Modelle werden sichtbar.

4 Literatur

- [1] Kolbe, P., Pfennigschmidt, S., Pahl, P.J.: *FATIMA Meilensteine 1-4*. (interne Berichte an die DeTeBerkom GmbH)
- [2] *ISO 10303 Product Data Representation and Exchange - Part 11: The EXPRESS Language Reference Manual, Part 21: Clear Text Encoding of the Exchange Structure, Part 22: Standard Data Access Interface*
- [3] Kim W. et al.: *Modern Database Systems - The Object Model, Interoperability and Beyond*. Addison Wesley, New York 1995
- [4] Ranglack D.: "Ein Typsystem für CAD-Systeme - Ein Integrationskonzept für computergestützte Planungsprozesse" in "Internationales Kolloquium über Anwendungen der Informatik und der Mathematik in Architektur und Bauwesen". Weimar, 1994